Two Sources of Control over the Generation of Software Instructions*

Anthony Hartley
Language Centre
University of Brighton, Falmer
Brighton BN1 9PH, UK
afh@itri.bton.ac.uk

Cécile Paris†
Information Technology Research Institute
University of Brighton, Lewes Road
Brighton BN2 4AT, UK
clp@itri.brighton.ac.uk

1 Introduction

Our work addresses the generation of software manuals in French and English, starting from a semantic model of the task to be documented (Paris et al., 1995). Our prime concern is to be able to exercise control over the mapping from the task model to the generated text. We set out to establish whether the task model alone is sufficient to control the linguistic output of a text generation system, or whether additional control is required. In this event, an obvious source to explore is the communicative purpose of the author, which is not necessarily constant throughout a manual. Indeed, in a typical software manual, it is possible to distinguish at least three sections, each with a different purpose: a tutorial containing exercises for new users, a series of step-by-step instructions for the major tasks to be accomplished, and a ready-reference summary of the commands.

We need, therefore, to characterise the linguistic expressions of the different elements of the task model, and to establish whether these expressions are sensitive or not to their context, that is, the functional section in which they appear. This paper presents the results of an analysis we conducted to this end on a corpus of software instructions in French.

2 Methodology

The methodology we employed is similar to that endorsed by (Biber, 1995). It is summarised as follows:

1. Collect the texts and note their situational characteristics. We consider two such characteristics: task structure and communicative purpose.

2. Identify the range of linguistic features to be included in the analysis;

3. Code the corpus in terms of the selected features;

4. Compute the frequency count of each linguistic feature;

5. Identify co-occurrences between linguistic features and the situational characteristics under consideration.

We first carried out a classical sublanguage analysis on our corpus as a whole, without differentiating between any of the situational characteristics (Hartley and Paris, 1995). This initial description was necessary to give us a clear statement of the linguistic potential required of our text generator, to which we could relate any restrictions on language imposed by situational variables. Thus we can account for language restrictions by appealing to general discourse principles, in keeping with the recommendations of (Kittredge, 1995) and (Biber, 1995) for the definition of sublanguages.

We then correlated task elements with grammatical features. Finally, where linguistic realisation was under-determined by task structure alone, we established whether the communicative purpose provided more discriminating control over the linguistic resources available.

3 Linguistic Framework: Systemic Functional Linguistics

Our analysis was carried out within the framework of Systemic-Functional Linguistics (SFL) (Halliday,

*This work is partially supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant J19221, by BC/DAAD ARC Project 293, by the Commission of the European Union Grant LRE-62009, and by the Office of Naval Research Grant N00014-96-1-0465.

†Starting this fall, Dr. Paris' address will be CSIRO, Division of Information Technology, Sydney Laboratory, Building E6B, Macquarie University Campus, North Ryde, Sydney, NSW 2113, Australia.

1978; Halliday, 1985) which views language as a resource for the creation of meaning. SFL stratifies meaning into context and language. The strata of the linguistic resources are organised into networks of choices, each choice resulting in a different meaning *realised* (i.e., expressed) by appropriate structures. The emphasis is on *paradigmatic* choices, as opposed to *syntagmatic* structures. Choices made in each stratum constrain the choices available in the stratum beneath. Context thus constrains language.

This framework was chosen for several reasons. First, the organisation of linguistic resources according to this principle is well-suited to natural language generation, where the starting point is necessarily a communicative goal, and the task is to find the most appropriate expression for the intended meaning (Matthiessen and Bateman, 1991). Second, a functional perspective offers an advantage for multilingual text generation, because of its ability to achieve a level of linguistic description which holds across languages more effectively than do structurally-based accounts. The approach has been shown capable of supporting the sharing of linguistic resources between languages as structurally distinct as English and Japanese (Bateman et al., 1991a; Bateman et al., 1991b). It is therefore reasonable to expect that at least the same degree of commonality of description is achievable between English and French within this framework. Finally, KPML (Bateman, 1994), the tactical generator we employ, is based on SFL, and it is thus appropriate for us to characterise the corpus in terms immediately applicable to our generator.

4 Coding features

Our lexico-grammatical coding was done using the networks and features of the Nigel grammar (Halliday, 1985). We focused on four main concerns, guided by previous work on instructional texts, e.g., (Lehrberger, 1986; Plum et al., 1990; Ghadessy, 1993; Kosseim and Lapalme, 1994).

- *Relations between processes*: to determine whether textual cohesion was achieved through conjunctives or through relations implicit in the task structure elements. Among the features considered were clause dependency and conjunction type.
- *Agency*: to see whether the actor performing or enabling a particular action is clearly identified, and whether the reader is explicitly addressed. We coded here for features such as voice and agent types.
- *Mood, modality and polarity*: to find out the extent to which actions are presented to the reader as being desirable, possible, mandatory, or prohibited. We coded for both true and implicit negatives, and for both personal and impersonal expressions of modality.
- *Process types*: to see how the domain is construed in terms of actions on the part of the user and the software. We coded for sub-categories of material, mental, verbal and relational processes.

5 The Corpus

The analysis was conducted on the French version of the Macintosh MacWrite manual (Kaehler, 1983). The manual is derived from an English source by a process of *adaptive translation* (Sager, 1993), i.e., one which localises the text to the expectations of the target readership. The fact that the translation is adaptive rather than literal gives us confidence in using this manual for our analysis.[1] Furthermore, we know that Macintosh documentation undergoes thorough local quality control. It certainly conforms to the principles of good documentation established by current research on technical documentation and on the needs of end-users, e.g., (Carroll, 1994; Hammond, 1994), in that it supplies clear and concise information for the task at hand. Finally, we have been assured by French users of the software that they consider this particular manual to be well written and to bear no unnatural trace of its origins.

Technical manuals within a specific domain constitute a sublanguage, e.g., (Kittredge, 1982; Sager et al., 1980). An important defining property of a sublanguage is that of closure, both lexical and syntactic. Lexical closure has been demonstrated by, for example, (Kittredge, 1987), who shows that after as few as the first 2000 words of a sublanguage text, the number of new word types increases little if at all. Other work, e.g., (Biber, 1988; Biber, 1989) and (Grishman and Kittredge, 1986) illustrates the property of syntactic closure, which means that generally available constructions just do not occur in this or

[1] We would have preferred to use a manual which originated in French to exclude all possibility of interference from a source language, but this proved impossible. Surprisingly, it appears that large French companies often have their documents authored in English by francophones and subsequently translated into French. One large French software house that we contacted does author its documentation in French, but had registered considerable customer dissatisfaction with its quality. We decided, therefore, that their material would be unsuitable for our purposes.

Goals: La sélection
Gloss: Selection

Pour sélectionner un mot, (faites un double-clic sur le mot)
Gloss: To select a word, (do a double-click on the word)

Functions: (Fermer –) Cet article permet de fermer une fenêtre activée
Gloss: (Close –) This command enables you to close the active window

Constraints: Si vous donnez à votre document le titre d'un document déjà existant, (une zone de dialogue apparaît)
Gloss: If you give your document the title of an existing document, (a dialog box appears)

Results: (Choisissez Coller dans le menu Edition –) Une copie du contenu du presse-papiers apparaît
Gloss: (Choose Paste from the Edit menu –) A copy of the content of the clipboard appears

Substeps: Fermez la fenêtre Rechercher
Gloss: Close the Find window

Ensuite, on ouvre le document de destination
Gloss: Next, one opens the target document

Figure 1: Examples of task element expressions

that sublanguage. In the light of these results, we considered a corpus of 15000 words to be adequate for our purposes, at least for an initial analysis.

The MacWrite manual is organised into three chapters, corresponding to the three different sections identified earlier: a tutorial, a series of step-by-step instructions for the major word-processing tasks, and a ready-reference summary of the commands. We omitted the tutorial because the generation of such text is not our concern, retaining the other two chapters which provide the user with generic instructions for performing relevant tasks, and descriptions of the commands available within MacWrite. The overlap in information between the two chapters offers opportunities to observe differences in the linguistic expressions of the same task structure elements in different contexts.

6 Task Structure

Task structure is constituted by five types of task elements, which we define below. We used the notion of task structure element both as a contextual feature for the analysis and to determine the segmentation of the text into units. Each unit is taken to be the expression of a single task element.

Our definition of the task elements is based on the concepts and relations commonly chosen to represent a task structure (a goal and its associated plan), e.g., (Fikes and Nilsson, 1971; Sacerdoti, 1977), and on related research, e.g., (Kosseim and Lapalme, 1994). Our generator produces instructions from an underlying semantic knowledge base which uses this representation (Paris et al., 1995). To generate an instruction for performing a task is to chose some task elements to be expressed and linearise them so that they form a coherent set for a given goal the user might have. We distinguish the following elements, and provide examples of them in Figure 1:[2]

goals: actions that users will adopt as goals and which motivate the use of a plan.

functions: actions that represent the functionality of an interface object (such as a menu item). A function is closely related to a goal, in that it is also an action that the user may want to per-

[2]The text in parentheses in the Figure is part of the linguistic context of the task element rather than the element itself.

form. However, the function is accessed through the interface object, and not through a plan.

constraints and preconditions: states which must hold before a plan can be employed successfully. The domain model distinguishes constraints (states which cannot be achieved through planning) and preconditions (states which can be achieved through planning). We do not make this distinction in the linguistic analysis and regroup these related task structure elements under one label. We decided to proceed in this way to determine at first how constraints in general are expressed. Moreover, it is not always clear *from the text* which type of constraint is expressed. Drawing too fine distinctions in the corpus analysis at this point, in the absence of a test for assigning a unit to one of these constraint types, would have rendered the results of the analysis more subjective and thus less reliable.

results: states which arise as planned or unplanned effects of carrying out a plan. While it might be important to separate planned and unplanned effects in the underlying representation, we again abstract over them in the lexicogrammatical coding.

sub-steps: actions which contribute to the execution of the plan. If the sub-steps are not primitive, they can themselves be achieved through other plans.

7 The Coding Procedure

No tools exist to automate a functional analysis of text, which makes coding a large body of text a time-consuming task. We first performed a detailed coding of units of texts on approximately 25% of the corpus, or about 400 units,[3] using the WAG coder (O'Donnell, 1995), a tool designed to facilitate a functional analysis.

We then used a public-domain concordance program, MonoConc (Barlow, 1994), to verify the representativeness of the results. We enumerated the realisations of those features that the first analysis had shown as marked, and produced KWIC[4] listings for each set of realisations. We found that the second analysis corroborated the results of the first, consistent with the nature of sublanguages.

[3]The authors followed guidelines for identifying task element units which had yielded consistent results when used by students coding other corpora.

[4]Key Word In Context

8 Distribution of Grammatical Features over Task Structure and Communicative Purpose

We examined the correlations between lexicogrammatical realisations and task elements and communicative purpose. The results are best expressed using tables generated by WAG: given any system, WAG splits the codings into a number of sets, one for each feature in that system. Percentages and means are computed, and the sets are compared statistically, using the standard T-test. WAG displays the results with an indicator of how statistically significant a value is compared to the combined means in the other sets. The counts were all done using the local mean, that is, the feature count is divided by the total number of codings which select that feature's system. Full definitions of the features can be found in (Halliday, 1985; Bateman et al., 1990).

In some cases, the type of task element is on its own sufficient to determine, or at least strongly constrain, its linguistic realisation. The limited space available here allows us to provide only a small number of examples, shown in Figure 2. We see that the use of modals is excluded in the expression of function, result and constraint, whereas goal and substep do admit modals. As far as the polarity system is concerned, negation is effectively ruled out for function, goal and substep. Finally, with respect to the mood system, only substep can be realised through imperatives.

In other cases, however, we observe a diversity of realisations. We highlight here three cases: modality in goal, polarity in constraint, and mood in substep. In such cases, we must appeal to another source of control over the apparently available choices. We have looked to the construct of *genre* (Martin, 1992) to provide this additional control, on two grounds: (1) since genres are distinguished by their communicative purposes, we can view each of the functional sections already identified as a distinct genre; (2) genre is presented as controlling text structure and realisation. In Martin's view, genre is defined as a staged, goal-oriented social process realised through register, the context of situation, which in turn is realised in language to achieve the goals of a text. Genre is responsible for the selection of a text structure in terms of task elements. As part of the realisation process, generic choices preselect a register associated with particular elements of text structure, which in turn preselect lexico-grammatical features. The coding of our text in terms genre and task elements thus allows us to establish the role played by genre in the realisations of the task elements. It

	Function	Result	Constraint	Goal	Substep
Modal-System					
modal	0%	1%	0%	24%	16%
non-modal	100%	99%	100%	76%	84%
polarity					
positive	100%	90%	68%	97%	97%
negative	0%	10%	32%	3%	3%
mood-system					
declarative	100%	100%	100%	100%	24%
imperative	0%	0%	0%	0%	76%

Figure 2: Selective realisations of task elements

	Ready-Reference	Procedure	Elaboration
Sub-step	37%	77%	42%
Goal	11%	23%	14%
Constraint	10%	0%	14%
Result	23%	0%	27%
Function	11%	0%	3%

Figure 3: Distribution of task structure elements over genres

will also allow us to determine the text structures appropriate in each genre, a study we are currently undertaking. This is consistent with other accounts of text structure for text generation in technical domains, e.g., (McKeown, 1985; Paris, 1993; Kittredge et al., 1991).

For those cases where the realisation remains under-determined by the task element type, we conducted a finer-grained analysis, by overlaying a genre partition on the undifferentiated data. We distinguished earlier two genres with which we are concerned: ready-reference and step-by-step. In the manual analysed, we recognised two more specific communicative purposes in the step-by-step section: to enable the reader to perform a task, and to increase the reader's knowledge about the task, the way to achieve it, or the properties of the system as a whole. Because of their distinct communicative purposes, we again feel justified in calling these genres. We label them respectively *procedure* and *elaboration*. The intention that the reader should recognise the differences in function of each section is underscored by the use of distinctive typographical devices, such as fonts and lay-out.[5]

[5]See (Hartley and Paris, 1995) for examples extracted from the manuals.

The first step at this stage of the analysis was to establish whether there was an effective overlap in task elements among the three genres under consideration. The results of this step is shown in Figure 3. Sub-step and goal are found in all three genres, while constraint, result and function occur in both ready-reference and elaboration but are absent from procedure.

The next step was to undertake a comparative analysis of the lexico-grammatical features found in the three genres. This analysis indicated that the language employed in these different sections of the text varies greatly. We summarise here the two genres that are strongly contrasted: procedure and ready-reference. Elaboration shares features with both of these.

procedure: The top-level goal of the user is expressed as a nominalisation. Actions to be achieved by the reader are almost exclusively realised by imperatives, directly addressing the reader. These actions are mostly material directed actions, and there are no causatives. Few modals are employed, and, when they are, it is to express obligation impersonally. The polarity of processes is always positive. Procedure employs mostly independent clauses, and, when

	Procedure	Ready-Reference	Elaboration
Non-modal	100.0%	75.0%	72.6%
Modal	0.0%	25.0%	28.4%

Figure 4: Genre-related differences in the modal system for goal

	Ready-Reference	Elaboration
Negative	0.0%	41.7%
Positive	100%	58.3%

Figure 5: Genre-related differences in the polarity system for constraint

	Procedure	Ready-Reference	Elaboration
Imperative	97.3%	44.4%	77.6%
Declarative	2.7%	55.6%	22.4%

Figure 6: Genre-related differences in the mood system for substep

clause complexes are used, the conjunctions are mostly purpose (linking a user goal and an action) and alternative (linking two user actions or two goals).

ready-reference: In this genre, all task elements are always realised through clauses. The declarative mood predominates, with few imperatives addressing the reader. Virtually all the causatives occur here. On the dimension of modality, the emphasis is on personal possibility, rather than obligation, and on inclination. We find in this genre most of the verbal processes, entirely absent from procedure. Ready-reference is more weighted than procedure towards dependent clauses, and is particularly marked by the presence of temporal conjunctions.

The analysis so far demonstrates that genre, like task structure, provides some measure of control over the linguistic resources but that neither of these alone is sufficient to drive a generation system. The final step was therefore to look at the realisations of the task elements differentiated by genre, in cases where the realisation was not strongly determined by the task element.

We refer the reader back to Figure 2, and the under-constrained cases of modality in goal, polarity in constraint, and mood in substep. Figure 4 shows the realisations the task element goal with respect to the modal system, which brings into sharp relief the absence of modality from procedure. Figure 5 presents the realisations by genre of the polarity system for constraint. We observe that only positive polarity occurs in ready-reference. Finally, we note from Figure 6 that the realisation of substeps is heavily loaded in favour of imperatives in procedure.

These figures show that genre does indeed provide useful additional control over the expression of task elements, which can be exploited by a text generation system. Neither task structure nor genre alone is sufficient to provide this control, but, taken together, they offer a real prospect of adequate control over the output of a text generator.

9 Related Work

The results from our linguistic analysis are consistent with other research on sublanguages in the instructions domain, in both French and English, e.g., (Kosseim and Lapalme, 1994; Paris and Scott, 1994). Our analysis goes beyond previous work by identifying within the discourse context the means for exercising explicit control over a text generator.

An interesting difference with respect to previous descriptions is the use of the true (or direct) imperative to express an action in the procedure genre, as results from (Paris and Scott, 1994) seem to indicate that the infinitive-form of the imperative is preferred in French. These results, however, were obtained from a corpus of instructions mostly for domestic appliances as opposed to software manuals.

Furthermore the use of the infinitive-form in instructions in general as observed by (Kocourek, 1982) is declining, as some of the conventions already common in English technical writing are being adopted by French technical writers, e.g., (Timbal-Duclaux, 1990).

We also note that the patterns of realisations uncovered in our analysis follow the principle of good technical writing practice known as the *minimalist approach*, e.g., (Carroll, 1994; Hammond, 1994). Moreover, we observe that our corpus does not exhibit shortcomings identified in a Systemic Functional analysis of English software manuals (Plum et al., 1990), such as a high incidence of agentless passive and a failure to distinguish the function of informing from that of instructing.

Other work has focused on the cross-linguistic realisations of two specific semantic relations (*generation* and *enablement*) (Delin et al., 1994; Delin et al., 1996), in a more general corpus of instructions for household appliances. Our work focuses on the single application domain of software instructions. However, it takes into consideration the whole task structure and looks at the realisation of semantic elements as found in the knowledge base, instead of two semantic relations not explicitly present in the underlying semantic model.

10 Conclusion

In this paper we have shown how genre and task structure provide two essential sources of control over the text generation process. Genre does so by constraining the selection of the task elements and the range of their expressions. These elements, which are the procedural representation of the user's tasks, constitute a layer of control which mediates between genre and text, but which, without genre, cannot control the grammar adequately.

The work presented here is informing the development of our text generator by specifying the necessary coverage of the French grammar to be implemented, the required discourse structures, and the mechanisms needed to control them. We continue to explore further situational and contextual factors which might allow a system to fully control its available linguistic resources.

References

[Barlow1994] Michael Barlow. 1994. *A Guide to MonoConc*. Athelston, Houston, TX.

[Bateman et al.1990] John A. Bateman, Robert T. Kasper, Johanna D. Moore, and Richard Whitney. 1990. A general organization of knowledge for natural language processing: The Penman Upper Model. Technical report, USC/ISI, March.

[Bateman et al.1991a] John A. Bateman, Christian M.I.M. Matthiessen, Keizo Nanri, and Licheng Zeng. 1991a. The re-use of linguistic resources across languages in multilingual generation components. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence, Volume 2, Sydney, Australia*, pages 966 – 971. Morgan Kaufmann Publishers.

[Bateman et al.1991b] John A. Bateman, Christian M.I.M. Matthiessen, Keizo Nanri, and Licheng Zeng. 1991b. Multilingual text generation: an architecture based on functional typology. In *International Conference on Current Issues in Computational Linguistics*, Penang, Malaysia.

[Bateman1994] John A. Bateman. 1994. KPML: The KOMET-Penman (Multilingual) Development Environment. Technical report, Institut für Integrierte Publikations- und Informationssysteme (IPSI), GMD, Darmstadt, September. Release 0.6.

[Biber1988] Douglas Biber. 1988. *Variation Across Speech and Writing*. Cambridge University Press, Cambridge UK.

[Biber1989] Douglas Biber. 1989. A typology of English texts. *Linguistics*, 27:3–43.

[Biber1995] Douglas Biber. 1995. *Dimensions of Register Variation: A Cross-linguistic comparison*. Cambridge University Press, Cambridge UK.

[Carroll1994] John Carroll. 1994. Techniques for minimalist documentation and user interface design. In *Quality of Technical Documentation*, pages 67–75. Rodopi, Amsterdam.

[Delin et al.1994] Judy Delin, Anthony Hartley, Cécile Paris, Donia Scott, and Keith Vander Linden. 1994. Expressing procedural relationships in multilingual instructions. In *Proceedings of the Seventh International Workshop on Natural Language Generation,* Kennebunkport, MN, 21–24 June 1994, pages 61–70.

[Delin et al.1996] Judy Delin, Donia Scott, and Anthony Hartley. 1996. Language-specific mappings from semantics to syntax. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, Copenhagen, Denmark, August.

[Fikes and Nilsson1971] R. E. Fikes and Nils Nilsson. 1971. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

[Ghadessy1993] Mohsen Ghadessy, editor. 1993. *Register Analysis: Theory and Practice*. Frances Pinter, London.

[Grishman and Kittredge1986] Ralph Grishman and Richard Kittredge, editors. 1986. *Analyzing Language in Restricted Domains*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[Halliday1978] Michael A. K. Halliday. 1978. *Language as a Social Semiotic: The Social Interpretation of Language and Meaning*. Edward Arnold, London.

[Halliday1985] Michael A. K. Halliday. 1985. *An Introduction to Functional Grammar*. Edward Arnold, London.

[Hammond1994] Nick Hammond. 1994. Less is More: The Minimalist Approach. Usability Now! A Guide to Usability. Published by the Open University and the Department of Trade and Industry.

[Hartley and Paris1995] Anthony Hartley and Cécile Paris. 1995. French corpus analysis and grammatical description. Technical Report Project IED/4/1/5827, ITRI, November.

[Kaehler1983] Carol Kaehler. 1983. *Macintosh MacWrite*. Apple Seedrin, Les Ulis, France.

[Kittredge et al.1991] Richard Kittredge, Tanya Korelsky, and Owen Rambow. 1991. On the Need for Domain Communication Knowledge. *Computational Intelligence*, 7(4):305–314, November.

[Kittredge1982] Richard Kittredge. 1982. Variation and Homogeneity of Sublanguages. In Richard Kittredge and J. Lehrberger, editors, *Sublanguage: Studies of language in restricted semantic domains*, pages 107–137. de Gruyter, Berlin and New York.

[Kittredge1987] Richard Kittredge. 1987. The significance of sublanguage for automatic translation. In Sergei Nirenburg, editor, *Machine Translation: Theoretical and methodological issues*, pages 59–67. Cambridge University Press, London.

[Kittredge1995] Richard Kittredge. 1995. Efficiency vs. Generality in Interlingual Design: Some Linguistic Considerations. In *the Working notes of the IJCAI-95 Workshop on Multilingual Text Generation,* August 20–21, Montréal, Canada.

[Kocourek1982] Rostislav Kocourek. 1982. *La langue française de la technique et de la science*. Brandstetter Verlag, Wiesbaden, Germany.

[Kosseim and Lapalme1994] Leila Kosseim and Guy Lapalme. 1994. Content and rhetorical status selection in instructional texts. In *Proceedings of the Seventh International Workshop on Natural Language Generation,* Kennebunkport, MN, 21–24 June 1994, pages 53–60.

[Lehrberger1986] John Lehrberger. 1986. Sublanguage Analysis. In Ralph Grishman and Richard Kittredge, editors, *Analyzing Language in Restricted Domains*, pages 19–38. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[Martin1992] James R. Martin. 1992. *English text: systems and structure*. Benjamins, Amsterdam.

[Matthiessen and Bateman1991] Christian M.I.M. Matthiessen and John A. Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Frances Pinter Publishers and St. Martin's Press, London and New York.

[McKeown1985] Kathleen R. McKeown. 1985. *Text Generation*. Cambridge University Press, New York.

[O'Donnell1995] Michael O'Donnell. 1995. From Corpus to Codings: Semi-Automating the Acquisition of Linguistic Features. Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, Stanford University, California, March 27 - 29, March.

[Paris and Scott1994] Cécile Paris and Donia Scott. 1994. Stylistic variation in multilingual instructions. In *Proceedings of the Seventh International Workshop on Natural Language Generation,* Kennebunkport, MN, 21–24 June 1994, pages 45 – 52.

[Paris et al.1995] Cécile Paris, Keith Vander Linden, Markus Fischer, Anthony Hartley, Lyn Pemberton, Richard Power, and Donia Scott. 1995. A support tool for writing multilingual instructions. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence,* August 20–25, Montréal, Canada, pages 1398–1404.

[Paris1993] Cécile L. Paris. 1993. *User Modelling in Text Generation*. Frances Pinter, London.

[Plum et al.1990] Guenter A. Plum, Christian M.I.M. Matthiessen, Michael A.K. Halliday, and Natalie Shea. 1990. The Electronic Discourse Analyzer Project: Report on Textual Analysis of FJ Manuals. Technical report, Fujitsu Australia Limited, Documentation Engineering Division. EDA Project Deliverables; Output 7.

[Sacerdoti1977] Earl D. Sacerdoti. 1977. *A Structure for Plans and Behavior*. Elsevier, New York.

[Sager et al.1980] Juan C. Sager, David Dungworth, and Peter F. McDonald. 1980. *English Special Languages*. Brandstetter Verlag, Wiesbaden, Germany.

[Sager1993] Juan C. Sager. 1993. *Language Engineering and Translation: Consequences of Automation*. John Benjamins Publishing Company, Amsterdam.

[Timbal-Duclaux1990] Louis Timbal-Duclaux. 1990. *La communication écrite scientifique et technique*. ESF, Paris, France.